1. Introduction
   1. Purpose

The SRS describes the functional and non-functional requirements for the initial release of StudySync. It is intended to guide the system's implementation and verify the web extension's correct functioning. All requirements are committed for the initial release.

* 1. Project scope

Please refer to section 2, “Vision and Scope”.

1. Overall description
   1. Product perspective

StudySync is a new web extension that aids Coursera students in their studies through a collection of productivity tools, including a whitelist web blocker and a time tracker dedicated to the Coursera platform. The following context diagram illustrates the initial release's external entities and system interfaces.

* 1. User classes and characteristics

Please refer to section 3, “User Requirements.”

* 1. Operating environment

OE-1: StudySync shall operate correctly on the following web browsers: Google Chrome (all versions).

* 1. Design and implementation constraints

CO-1: StudySync’s design, code and maintenance documentation shall conform to the *Google Webstore Program Policies,* [*https://developer.chrome.com/docs/webstore/program-policies*](https://developer.chrome.com/docs/webstore/program-policies)*.*

CO-2: StudySync’s access to Coursera shall conform to *Coursera’s Terms of Use,* [*https://www.coursera.org/about/terms*](https://www.coursera.org/about/terms)*.*

* 1. Assumptions and dependencies

AS-1: The user is a student enrolled in the BSc in Computer Science offered on Coursera in partnership with the University of London.

AS-2: The user will enter valid URLs into the whitelist form.

DE-1: The user has allowed StudySync to access their website data.

3 System features

3.1. Interact with the Extension via the Dropdown Menu

3.1.1. Description

The user can interact with the extension via the dropdown menu. This will allow users to quickly toggle the whitelist web blocker and view their current time tracker stats. The dropdown will also enable users to navigate to the HTML pages provided by the extension.

3.1.2 Functional Requirements

|  |  |
| --- | --- |
| Dropdown.Whitelist: | **Dropdown whitelist functionality** |
| .Toggle: | StudySync shall toggle the whitelist between on and off when the user clicks the toggle button. |
| .AddToWhitelist: | StudySync shall add the current domain name to the whitelist. |
| .Reload: | StudySync shall reload the current webpage. |
| .EditWhitelistButton: | StudySync shall redirect the user to the whitelist HTML page when the user clicks this button. |

|  |  |
| --- | --- |
| Dropdown.TimeTracker: | **Dropdown time-tracker functionality** |
| .Display: | StudySync shall display the current course and task being tracked along with the total time spent on the current task. |
| .StatsButton: | StudySync shall redirect the user to the data dashboard HTML page when the user clicks this button. |
| .SettingsButton: | StudySync shall redirect the user to the time-tracker settings HTML page when the user clicks this button. |

3.2 Navigate the HTML Pages via the Navbar

3.2.1 Description

The user can navigate the extension HTML pages via a navbar, directing the user to the associated HTML page.

3.2.2 Functional requirements

|  |  |
| --- | --- |
| Navbar: | **HTML pages navbar functionality** |
| .Whitelist: | When the user clicks on the whitelist navigation link, StudySync shall direct the user to the whitelist form page. |
| .DataDashboard: | When the user clicks on the data dashboard navigation link, StudySync shall direct the user to the data dashboard page. |
| .TimeTrackerSettings: | When the user clicks on the time-tracker settings navigation link, StudySync shall direct the user to the time-tracker settings page. |

3.3 Edit the Whitelist Form

3.3.1 Description

The user can adjust the URLs to be allowed by the whitelist web blocker. They can add and remove URLs via the form and save the updated list by clicking the save button.

3.3.2 Functional Requirements

|  |  |
| --- | --- |
| Whitelist.Form: | **Editing whitelist form** |
| .AddURL: | StudySync shall allow the user to enter a new URL into the whitelist form. |
| .RemoveURL: | StudySync shall allow the user to remove a saved URL. |
| .SaveButton: | When the user clicks the save button, StudySync shall save the current whitelist form. |

3.4 Whitelist all Attempted URL Queries

3.4.1 Description

StudySync will automatically verify that all URL requests made by the web browser are allowed to reach the user. The extension will verify the domain name against the whitelist form.

3.4.2 Functional requirements

|  |  |
| --- | --- |
| Whitelist.Verify: | **Verify user can access a given URL** |
| .RetrieveURL: | StudySync shall retrieve all URL queries made by the web browser. |
| .CompareURL: | StudySync shall compare all URL query domain names made by the web browser with the whitelist. |
| .Allow: | If the Domain name is on the whitelist, StudySync shall allow access to the website. |
| .Deny: | If the Domain name is not on the whitelist, StudySync shall not allow access to the website. |
| .Alert: | StudySync shall alert the user via a custom URL page indicating the requested URL has been blocked. |

3.5 Track Coursera Study Time

3.5.1 Description

When a user has a URL open with the domain name coursera.org, StudySync will automatically track the time the user spends studying a course on the Coursera platform. The extension will track the time spent on the course and the current task.

3.5.2 Functional requirements

|  |  |
| --- | --- |
| TimeTracker.Track: | **Track time spent studying on Coursera** |
| .Validate: | StudySync shall validate the domain name of all URLs to check if the user is on a Coursera domain. |
| .CheckCourse: | StudySync shall confirm if the full URL is related to a course selected by the user in the time tracker settings. |
| .TrackCourse: | If the user is currently in a course Coursera section. StudySync shall start tracking time spent in the course. |
| .TrackTask: | If the user is within a given task, then StudySync shall track the time spent on the current task shown by the URL. This will be tracked under the current course the user is in. |
| .StoreData: | StudySync shall automatically store the time-tracker data locally in a flat format. |

3.6 View Time Tracker Stats on Data Dashboard

3.6.1 Description

The user can gain insights into their Coursera study habits via a data dashboard to visualize the time-tracker data. The user can break the data down by the overall semester, courses, and further details to be determined based on user feedback.

3.6.2 Functional requirements

|  |  |
| --- | --- |
| TimeTracker.DataDashboard: | **Display the data collected by the time tracker.** |
| .Group: | StudySync shall organize the data into logical separations. |
| .Semester: | StudySync shall display data visualizations based on the current semester's time-tracking data. |
| .Course: | StudySync shall display data visualizations based on the currently selected course. |
| .SelectCourse | StudySync shall allow the user to select the current course to be analyzed on the data dashboard. |
| .TBD: | User feedback can guide further visualizations when this is implemented. |
| .StoreData: | StudySync shall automatically store the time-tracker data locally in a flat format. |
| .Export: | StudySync shall allow the user to export the time-tracker data to CSV format. |

3.7 Adjust Time-Tracker Settings

3.7.1 Description

The user will need to adjust specific requirements and information for the time-tracker. While we know the user will need the option to start a new semester, this system requirement must be fleshed out during the sprint.

3.7.2 Functional requirements

|  |  |
| --- | --- |
| TimeTracker.Settings: | **Adjust the time-tracker settings** |
| .StartNewSemester: | When the user clicks this button, StudySync shall start a new tracking period by deleting old data. |
| .TBD: | The settings page will need to be fleshed out during the sprint related to it. User feedback will need to be gathered to ensure the time-tracker is meeting their needs. |

3.× System feature X

3.x.1 Description

3..2 Functional requirements

1. Data requirements
   1. Logical data model
   2. Data dictionary
   3. Reports
   4. Data acquisition, integrity, retention, and disposal
2. External interface requirements
   1. User interfaces
   2. Software interfaces
   3. Hardware interfaces
   4. Communications interfaces
3. Quality attributes
   1. Usability
   2. Performance
   3. Security
   4. Safety
   5. [others]

7. Internationalization and localization requirements

8. Other requirements

Appendix A: Glossary

Appendix B: Analysis Models